AMENDMENTS TO THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method of providing an open development kit, the open development kit having a virtual CPU adapted to execute a programmable logic controller (PLC) application program in a real-time operating environment and adapted to execute one or more extensions, the one or more extensions providing access into the scan cycle of the PLC and replaceable functionality to the operation of the PLC, the method comprising the steps of:

checking whether there are requests for application execution in the PLC;

sending a specific formatted load application request message from the PLC ~~a programmable logic controller (PLC)~~ to an open development kit (ODK) subsystem;

converting at the ODK subsystem the specific formatted load application request message to a generic formatted load application request message;

sending the generic formatted load ~~application~~ request message from the ODK subsystem to an application for requesting an extension to be loaded by the application; ~~and~~

requesting execution of the application, the application comprising an Application Wizard, the Application Wizard adapted to provide a first mode and a second mode,

checking whether any requests for information are waiting in the application and, if so, requesting information from the PLC by the application;

executing a function in the ODK subsystem specified by the application;

performing a task in the PLC associated with the executed function in the ODK subsystem;

returning a specific formatted response from the PLC to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK RTX Proxy DLL usable in the deterministic environment, the ODK RTX Proxy DLL adapted to communicated through shared memory with an ODK SB Add-ion DLL in the non-deterministic environment, wherein, in the first mode, the Application Wizard generates ~~adapted to generate~~ a Real Time Dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates ~~adapted to generated~~ a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment.

2. (Original) The method of claim 1, further comprising the steps of:

initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and

initializing the extension after the extension is loaded.

3. (Original) The method of claim 1, wherein the extension is a dynamic load library (DLL).

4. (Original) The method of claim 1, further comprising the step of checking whether a stop to run transition has occurred in the PLC and if so, sending a specific formatted activate application message from the PLC to the ODK subsystem.

5. (Currently Amended) The method of claim 4, further comprising the step of calling an activate function in the application by the ODK subsystem thereby permitting scan cycle execution, the Real Time Dynamic Link Library project comprising a helper class adapted to access data formats of Step 7 type data from a buffer that is passed between a central processing unit and the application.

6. (Cancelled)

7. (Original) The method of claim 6, further comprising the step of executing the generic execution request by the application.

8. (Original) The method of claim 6, further comprising the steps of:

sending a generic response from the application to the ODK subsystem;
converting the generic response to a specific formatted response; and
sending the specific formatted response from the ODK subsystem to the PLC.

9. (Original) The method of claim 8, further comprising the step of returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC.

10. (Cancelled)

11. (Original) The method of claim 10, wherein in the requesting information step the application uses a call back pointer to generically request information and the executing step executes the function in the ODK subsystem corresponding to the callback pointer.

12. (Original) The method of claim 11, in the executing a function step, the function is provided by a dynamic link library.

13. (Cancelled)

14. (Original) The method of claim 10, wherein when the checking determines that there are no requests for information waiting, and further comprising:

waiting until the PLC transitions from a run state to a stop state;
sending a deactivate request from the PLC to the ODK subsystem; and
calling a deactivate function in the application.

15. (Original) The method of claim 14, wherein when a memory clear or PLC shutdown occurs, the step of calling a release function in the application and unloading the extension occurs.

16. (Previously Presented) The method of claim 1, wherein in the sending a load application request from the PLC is from at least one of a soft PLC, a slot PLC and a hard PLC.

17. (Currently Amended) A system for open development that comprises a computer-readable medium encoded with a computer program, the computer program comprising:

one or more extensions adapted to provide real-time services ~~for use~~ in a real-time operating environment; and

a virtual CPU adapted to execute a programmable logic controller (PLC) application program in the real-time operating environment and adapted to

check whether there are requests for application execution in the PLC and, if so, receive a specific formatted execution request from the PLC, convert the specific formatted execution request to a generic execution request, send the generic execution request to the application, and execute the one or more extensions, the system comprising an Application Wizard, the Application Wizard adapted to provide a first mode and a second mode, and

check whether any requests for information are waiting in the application and, if so, request information from the PLC, execute a specified by the application, initiate performance of a task in the PLC associated with the executed function, receive a specific formatted response from the PLC, convert the specific formatted response to a generic response, and return the generic response to the application using a RTX Proxy DLL, the RTX Proxy DLL usable in the deterministic environment and adapted to communicated through shared memory in the non-deterministic environment,

wherein, in the first mode, the Application Wizard generates ~~adapted to generated~~ a Real Time Dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates ~~adapted to generate~~ a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

wherein the one or more extensions provide access into the scan cycle of the PLC and to provide replaceable functionality to the operation of the PLC, and

the deterministic environment is a WinAC RTX environment.

18. (Original) The system of claim 17, further comprising:

a system block loader adapted to load system blocks, the system blocks including at least one of a system function block, a system function, a system data block; and

an ODK SB Add-on dynamic link library (DLL)for implementing a common object module (COM) interface for the virtual CPU and system block loader.

19. (Original) The system of claim 17, wherein the one or more extensions are dynamic link libraries.

20. (Currently Amended) A system that comprises a computer-readable medium encoded with a computer program, the computer program adapted to provide an open development kit, the computer program comprising:

a means for executing a programmable logic controller (PLC) application program in the real-time operating environment and adapted to execute the one or more extensions in virtual CPU;

a means for determining whether there are requests for application execution in the PLC;

a means for sending a specific format load application request message from a programmable logic controller (PLC) to an open development kit (ODK) subsystem, in response to determining requests for application execution are in the PLC;

a means for converting the specific format load application request to a generic format load ~~application~~ request message by the ODK subsystem;

a means for sending the generic format load ~~application~~ request message from the ODK subsystem to an application requesting an extension to be loaded by the application; ~~and~~

a means for requesting execution of the application, the application comprising an Application Wizard, the Application Wizard adapted to provide a first mode and a second mode;[[,]]

a means for determining whether any requests for information are waiting in the application and, if so, requesting information from the PLC by the application;

a means for executing a function in the ODK subsystem specified by the application;

a means for performing a task in the PLC associated with the executed function in the ODK subsystem;

a means for returning a specific formatted response from the PLC to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

a means for returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK RTX Proxy DLL usable in the deterministic environment, the ODK RTX Proxy DLL adapted to communicated through shared memory with an ODK SB Add-ion DLL in the non-deterministic environment,

wherein, in the first mode, the Application Wizard generates ~~adapted to generated~~ a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates ~~adapted to generate~~ a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment..

21. (Original) The system of claim 20, further comprising the steps of:

a means for initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and

a means for initializing the extension after the extension is loaded.

22. (Original) The system of claim 20, further comprising a means for checking whether a stop to run transition has occurred in the PLC and if so, a means for sending a specific activate application message from the PLC to the ODK subsystem.

23. (Original) The system of claim 22, further comprising a means for calling an activate function in the application by the ODK subsystem thereby permitting scan cycle execution.

24. (Original) The system of claim 20, further comprising:

a means for checking whether there are requests for application execution in the PLC;

a means for sending a specific formatted execution request from the PLC to the ODK subsystem;

a means for converting the specific formatted execution request to a generic formatted execution request; and

a means for sending the generic formatted execution request from the ODK subsystem to the application.

25. (Original) The system of claim 24, further comprising a means for executing the generic formatted execution request by the application.

26. (Original) The system of claim 25, further comprising:

a means for sending a generic response from the application to the ODK subsystem;

a means for converting the generic response to a specific formatted response; and

a means for sending the specific formatted response from the ODK subsystem to the PLC.

27. (Original) The system of claim 26, further including a means for returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC.

28. (Original) The system of claim 20, further comprising:

a means for checking whether any requests for information are waiting in the application;

a means for requesting information from the PLC by the application;

a means for executing a function in the ODK subsystem specified by the application; and

a means for performing a task in the PLC associated with the executed function in the ODK subsystem.

29. (Original) The system of claim 28, wherein the means for requesting information by the application uses a call back pointer to request information and the means for executing the function in the ODK subsystem executes the function in the ODK subsystem corresponding to the callback pointer.

30. (Original) The system of claim 29, wherein the function is provided by a dynamic link library (DLL).

31. (Original) The system of claim 20, wherein the PLC is at least one of a soft PLC, a hard PLC, and a slot PLC.

32. (Original) The system of claim 20, further comprising a means for returning a specific formatted response from the PLC to the ODK subsystem, a means for converting the specific formatted response to a generic formatted response, and a means for returning the generic formatted response from the ODK subsystem to the application.

33. (Currently Amended) A computer-readable medium encoded with a computer program, the computer program comprising a kit for open development, the kit for open development comprising:

a means for executing a programmable logic controller (PLC) application program in a real-time operating environment adapted to execute the one or more extensions in a virtual CPU, a means for determining whether there are requests for application execution in the PLC,

a means for receiving a specific formatted message from a programmable logic controller (PLC), in response to determining requests for application execution are in the PLC;

a means for converting the specific formatted message to a generic formatted message; ~~and~~

a means for sending the generic formatted message to an application for execution, the application comprising an Application Wizard, the Application Wizard adapted to provide a first mode and a second mode;[[,]]

a means for checking whether any requests for information are waiting in the application;

a means for requesting information from the PLC by the application in response to determine requests for information are waiting in the application;

a means for executing a function in the ODK subsystem specified by the application;

a means for performing a task in the PLC associated with the executed function in the ODK subsystem;

a means for returning a specific formatted response from the PLC to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

a means for returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK RTX Proxy DLL usable in the deterministic environment, the ODK RTX Proxy DLL adapted to communicated through shared memory with an ODK SB Add-ion DLL in the non-deterministic environment,

wherein, in the first mode, the Application Wizard generates ~~adapted to generated~~ a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates ~~adapted to generate~~ a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinRTX environment.

34. (Previously Presented) The kit of claim 33, further comprising:

a means for receiving the generic formatted message from the application;

a means for converting the generic formatted message to the specific formatted message; and

a means for sending the specific formatted message to the PLC.

35. (Original) The kit of claim 34, wherein at least one of the means includes using a dynamic link library that loads replaceable functionality.

36. (Original) The kit of claim 34, wherein the generic formatted message is one of a response message and a request for information message.

37. (Original) The kit of claim 34, wherein the specific formatted message is sent to at least one of a soft PLC, a hard PLC and a slot PLC.

38. (Currently Amended) A computer program product comprising a computer ~~usable~~ storage medium having readable program code embodied in the medium, the computer program product includes:

a first computer program code to check whether there are requests for application execution in the PLC;

a second computer program code to send a specific formatted load application request message from a programmable logic controller (PLC) to an open development kit (ODK) subsystem;

a third ~~second~~ computer program code to convert the specific formatted load application request message to a generic formatted load ~~application~~ request message;

a fourth ~~third~~ computer program code to send the generic formatted load ~~application~~ request message from the ODK subsystem to an application for requesting an extension to be loaded by the application; ~~and~~

a fifth ~~fourth~~ computer program code to request execution of the application, the application comprising an Application Wizard, the Application Wizard adapted to provide a first mode and a second mode;[[,]]

a sixth computer program code to check whether any requests for information are waiting in the application and, if so, request information from the PLC by the application, execute a function in the ODK subsystem specified by the application, and perform a task in the PLC associated with the executed function in the ODK subsystem;

a seventh computer program code to return a specific formatted response from the PLC to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response, and return the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK RTX Proxy DLL usable in the deterministic environment, the ODK RTX Proxy DLL adapted to communicated through shared memory with an ODK SB Add-ion DLL in the non-deterministic environment; and

an eight computer program code to provide a virtual CPU adapted to execute a programmable logic controller (PLC) application program in the real-time operating environment and adapted to execute the one or more extensions,

wherein, in the first mode, the Application Wizard generates ~~adapted to generated~~ a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates ~~adapted to generate~~ a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment.